



## Sõbrad

Me ehitame sotsiaalvõrgustiku  $n$  inimesest numbritega  $0, \dots, n - 1$ . Mõned inimeste paarid võrgustikus on omavahel sõbrad. Kui isik  $x$  saab isiku  $y$  sõbraks, siis ka isik  $y$  saab isiku  $x$  sõbraks.

Inimesi lisatakse võrgustikku  $n$  etapis, millel on samuti numbrid  $0$  kuni  $n - 1$ . Isik  $i$  lisatakse etapis  $i$ . Etapis  $0$  lisatakse isik  $0$ , kes on sotsiaalvõrgu ainus liige. Igas järgmises  $n - 1$  etapis lisab uue isiku *kutsuja*, kes võib olla ükskõik kes sotsiaalvõrgustiku olemasolevatest liikmetest. Etapis  $i$  ( $0 < i < n$ ) saab selle etapi *kutsuja* lisada isiku  $i$  võrgustikku, kasutades üht järgmistest kolmest protokollist:

- *MinaOlenSinuSõber* teeb isiku  $i$  ainult *kutsujaga* sõbraks.
- *MinuSõbradOnSinuSõbrad* teeb isiku  $i$  sõbraks *kõigiga*, kes on sel hetkel *kutsuja* sõbrad. Pane tähele, et see *ei tee* isikut  $i$  *kutsuja* enda sõbraks.
- *MeOlemeSinuSõbrad* teeb isiku  $i$  sõbraks *kutsujaga* ning samuti *kõigiga*, kes on sel hetkel *kutsuja* sõbrad.

Pärast võrgustiku ehitamist on meil vaja leida *valim*, mille põhjal teha küsitlust, s.t valida võrgustikust grupp inimesi. Kuna sõpradel on sarnased huvid, ei tohiks *valim* sisaldada inimesi, kes on üksteise sõbrad. Igal isikul on ka *usaldusväärsus*, mida väljendatakse positiivse täisarvuga, ning me soovime leida maksimaalse summaarse usaldusväärसusega *valimi*.

## Näide

e tapp	kutsuja	protokoll	lisatud sõbrasuhted
1	0	MinaOlenSinuSõber	(1, 0)
2	0	MinuSõbradOnSinuSõbrad	(2, 1)
3	1	MeOlemeSinuSõbrad	(3, 1), (3, 0), (3, 2)
4	2	MinuSõbradOnSinuSõbrad	(4, 1), (4, 3)
5	0	MinaOlenSinuSõber	(5, 0)

Alguses on võrgustikus ainult isik  $0$ .

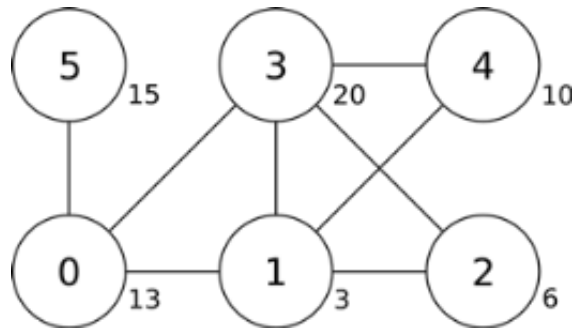
Etapi 1 *kutsuja* (isik  $0$ ) kutsub uue isiku  $1$  kasutades *MinaOlenSinuSõber* protokoll, seega saavad nad sõpradeks.

Etapi 2 *kutsuja* (jälle isik  $0$ ) kutsub isiku  $2$  kasutades *MinuSõbradOnSinuSõbrad*, mis teeb isiku  $1$  (*kutsuja* ainsa sõbra) isiku  $2$  ainsaks sõbraks.

Etapi 3 *kutsuja* (isik  $1$ ) lisab isiku  $3$ , kasutades *MeOlemeSinuSõbrad*, mis teeb isiku  $3$  sõbraks isikuga  $1$  (*kutsuja*) ning samuti isikutega  $0$  ja  $2$  (*kutsuja* sõbrad).

Etapid 4 ja 5 on näidatud eeltoodud tabelis.

Lõplik võrgustik on toodud järgmisel joonisel, kus arvud ringide sees tähistavad inimesi ning arvud ringide kõrval tähistavad usaldusväärсust. *Valim*, mis sisaldab isikuid  $3$  ja  $5$  annab usaldusväärсuse  $20 + 15 = 35$ , mis on maksimaalne summaarne usaldusväärсus.



## Ülesanne

Antud on iga etapi kirjeldus ning iga isiku usaldusväärsus. Leida suurima summaarse usaldusväärsega valim. Realiseerida tuleb ainult funktsioon `findSample`.

- `findSample(n, confidence, host, protocol)`
  - `n`: inimeste arv.
  - `confidence`: massiiv pikkusega `n`; `confidence[i]` on isiku `i` usaldusväärsus.
  - `host`: massiiv pikkusega `n`; `host[i]` on etapi `i` kutsuja.
  - `protocol`: massiiv pikkusega `n`; `protocol[i]` tähistab etapis `i` kasutatavat protokollit ( $0 < i < n$ ): 0 tähistab `MinaOlenSinuSõber`, 1 tähistab `MinuSõbradOnSinuSõbrad`, ja 2 tähistab `MeOlemeSinuSõbrad`.
  - Kuna esimeses etapis ei ole kutsujat, siis `host[0]` ja `protocol[0]` ei ole defineeritud ja programm ei tohiks neid kasutada.
  - Funktsioon peab tagastama valimi maksimaalse summaarse usaldusväärsega.

## Alamülesanded

Mõned alamülesanded kasutavad ainult osasid protokolle, nagu näidatud alltoodud tabelis.

alamülesanne	punkte	$n$	usaldusväärsus	protokollid
1	11	$2 \leq n \leq 10$	$1 \leq \text{usaldusväärsus} \leq 1\,000\,000$	Kõik kolm
2	8	$2 \leq n \leq 1000$	$1 \leq \text{usaldusväärsus} \leq 1\,000\,000$	Ainult <code>MinuSõbradOnSinuSõbrad</code>
3	8	$2 \leq n \leq 1000$	$1 \leq \text{usaldusväärsus} \leq 1\,000\,000$	Ainult <code>MeOlemeSinuSõbrad</code>
4	19	$2 \leq n \leq 1000$	$1 \leq \text{usaldusväärsus} \leq 1\,000\,000$	Ainult <code>MinaOlenSinuSõber</code>
5	23	$2 \leq n \leq 1000$	Kõik usaldusväärssused on 1	<code>MinuSõbradOnSinuSõbrad</code> ja <code>MinaOlenSinuSõber</code>
6	31	$2 \leq n \leq 100\,000$	$1 \leq \text{usaldusväärsus} \leq 10\,000$	Kõik kolm

## Realisatsiooni detailid

Esitada tuleb täpselt üks fail nimega `friend.c`, `friend.cpp` või `friend.pas`. Selles failis peab olema eelpool kirjeldatud alamprogramm järgmise signatuuriga. C/C++ programmis tuleb kaasata ka päisfail `friend.h`.

### C/C++ programm

```
int findSample(int n, int confidence[], int host[], int protocol[]);
```

### Pascal programm

```
function findSample(n: longint, confidence: array of longint, host: array  
of longint; protocol: array of longint): longint;
```

### Näidishindaja

Näidishindaja loeb sisendit järgmises formaadis:

- Esimesel real: `n`
- Teisel real: `confidence[0], ..., confidence[n-1]`
- Kolmandal real: `host[1], protocol[1], host[2], protocol[2], ..., host[n-1], protocol[n-1]`

Näidishindaja väljastab funktsiooni `findSample` tagastatud väärtuse.